

Diseño de Funciones Heurísticas

Stalin Muñoz Gutiérrez

Centro de Ciencias de la Complejidad
Universidad Nacional Autónoma de México (UNAM)

Las funciones heurísticas son la base de operación de los algoritmos de búsqueda informada. Su diseño y características tiene un impacto sobre la optimalidad de la solución y el desempeño en tiempo y memoria de los algoritmos.

¿De dónde vienen las heurísticas?

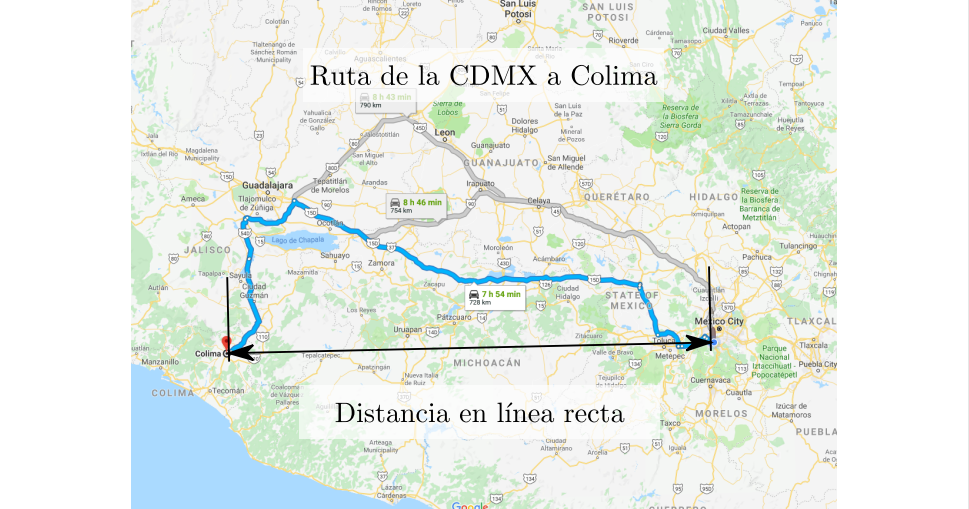
Quitando restricciones del problema original

Una manera común de proponer funciones heurísticas es quitar restricciones al problema original. Esto es, estimar el costo a la meta imaginando que el agente puede realizar acciones que el problema original no permite.

¿De dónde vienen las heurísticas?

En muchas ocasiones se puede proponer funciones heurísticas relajando las restricciones del problema original.

Distancia euclídeana en mapas



En el ejemplo de encontrar rutas en un mapa. Consideramos la distancia en línea recta como una buena heurística.

En este mapa vemos que la restricción para un vehículo es moverse sobre el sistema carretero. Esa ruta esta alejada de la línea recta.

Si quitamos esta restricción, estamos considerando que el agente

puede viajar en línea recta. A pesar de que esto no es cierto en la realidad, esta heurística es una buena aproximación respecto del costo a la meta.

Cumple además con el criterio de que es fácilmente computable.

Como en el mejor de los casos la carretera es una recta entre los puntos de interés, en un espacio Euclideo esta heurística es consistente

Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

¿Podemos diseñar una heurística a partir de relajar las restricciones del problema del rompecabezas del 8?

Distancia de Manhattan en el juego del 8



Inicial



Final

¿Podemos diseñar una heurística a partir de relajar las restricciones del problema del rompecabezas del 8?

Otro ejemplo donde podemos proponer una función heurística a partir de relajar las restricciones del problema es el rompecabezas del 8.

Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

Las restricciones del juego sólo permiten mover las fichas 7 y 8 en el estado inicial.

Distancia de Manhattan en el juego del 8



Inicial



Final

Las restricciones del juego sólo permiten mover las fichas 7 y 8 en el estado inicial.

La restricción del juego solo permite el movimiento de las fichas adyacentes al espacio libre.

En esta configuración las fichas 7 y 8.

Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

Las fichas fuera de lugar
ahora pueden moverse
vertical u horizontalmente.

Distancia de Manhattan en el juego del 8



Inicial



Final

Las fichas fuera de lugar
ahora pueden moverse
vertical u horizontalmente.

Al quitar esta restricción podemos considerar que las fichas se pueden mover no importando si estan adyacentes al espacio o no.

Aquí, para llegar a la meta las 6 fichas se podrían mover.

Distancia de Manhattan en el juego del 8

	1	3
4	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

Por ejemplo la ficha 4
debe moverse una casilla
hacia abajo.

└─ Distancia de Manhattan en el juego del 8



Inicial



Final

Por ejemplo la ficha 4
debe moverse una casilla
hacia abajo.

Por ejemplo.

La ficha 4 debe moverse una casilla hacia abajo para llegar a su posición deseada en la configuración final.

Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

Podemos usar como heurística el número de movimientos necesarios para llegar a la meta.

Distancia de Manhattan en el juego del 8



Inicial



Final

Podemos usar como heurística el número de movimientos necesarios para llegar a la meta.

La heurística entonces se define como el número de movimientos *sin restricción* necesarios para alcanzar la meta.

Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

$$\#_{\text{movimientos}}(4) = 1$$

└ Distancia de Manhattan en el juego del 8



Inicial



Final

$$\#_{\text{movimientos}}(4) = 1$$

Para la ficha 4 es un movimiento.

Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

$$D_{\text{Manhattan}}(\text{Inicial}, \text{Final}) = 1 + 1 +$$

$$\#_{\text{movimientos}}(1) = 1$$

└ Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

1	2	3
4	5	6
7	8	

Inicial

Final

$D_{\text{Manhattan}}(\text{Inicial}, \text{Final}) = 1 + 1 +$

$\#_{\text{movimientos}}(1) = 1$

Para la ficha 1, también es un movimiento.

Acumulamos, los movimientos de todas las fichas para calcular una distancia del estado inicial al final.

Esta distancia que considera movimientos únicamente horizontales y verticales, pero no diagonales se conoce como distancia de *Manhattan*.

Distancia de Manhattan en el juego del 8

4	1	3
7	2	5
	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

$$D_{\text{Manhattan}}(\text{Inicial}, \text{Final}) = 6$$

La distancia de Manhattan es consistente, y se puede calcular de manera eficiente.

Distancia de Manhattan en el juego del 8



Inicial



Final

$$D_{\text{Manhattan}}(\text{Inicial}, \text{Final}) = 6$$

La distancia de Manhattan es consistente, y se puede calcular de manera eficiente.

En nuestro ejemplo la distancia es 6.

La distancia de Manhattan es una heurística que no sobreestima.

Adicionalmente cumple con el requerimiento de que es computacionalmente poco costosa de calcular.

Distancia de Manhattan en el juego del 8

4	1	3
2		5
7	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

La distancia de Manhattan es una buena heurística, pero es más costosa que contar las fichas fuera de lugar.

Distancia de Manhattan en el juego del 8



Inicial



Final

La distancia de Manhattan es una buena heurística, pero es más costosa que contar las fichas fuera de lugar.

Sin embargo, es más costosa de calcular que simplemente contar las fichas fuera de lugar.

¿Nos conviene este extra costo?

Distancia de Manhattan en el juego del 8

4	1	3
2		5
7	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

Contar fichas fuera de lugar: h_1

Distancia de Manhattan: h_2

Distancia de Manhattan en el juego del 8



Inicial



Final

Contar fichas fuera de lugar: h_1
Distancia de Manhattan: h_2

Denotemos como h_1 la heurística que cuenta el número de fichas fuera de lugar.

Y como h_2 la heurística que calcula la distancia de Manhattan a la meta.

Distancia de Manhattan en el juego del 8

4	1	3
2		5
7	8	6

Inicial

$$h_1(\text{Inicial}, \text{Final}) = 1 + 1 + 1 + 1 + 1 = 5$$

$$h_2(\text{Inicial}, \text{Final}) = 1 + 1 + 2 + 1 + 1 = 6$$

1	2	3
4	5	6
7	8	

Final

└ Distancia de Manhattan en el juego del 8



Inicial



Final

$$h_1(\text{Inicial}, \text{Final}) = 1 + 1 + 1 + 1 + 1 = 5$$

$$h_2(\text{Inicial}, \text{Final}) = 1 + 1 + 2 + 1 + 1 = 6$$

Para la configuración mostrada
el valor de h_1 es 5.

El valor de h_2 es 6

Distancia de Manhattan en el juego del 8

4	1	3
2		5
7	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

$$h_1(\text{Inicial}, \text{Final}) = 1 + 1 + 1 + 1 + 1 = 5$$

$$h_2(\text{Inicial}, \text{Final}) = 1 + 1 + 2 + 1 + 1 = 6$$

Las heurísticas estiman dos valores distintos.

¿cuál es mejor?

└ Distancia de Manhattan en el juego del 8



Podemos ver que la diferencia radica en que la ficha 2 tiene que moverse dos veces para llegar a la configuración deseada. Una a la derecha y otra hacia arriba.

Si las heurísticas dan valores diferentes para la misma configuración.
 ¿cuál es mejor?

Para responder a esta pregunta no solo hay que tomar en cuenta el costo de calcular la heurística, sino también que tan buena es la estimación de la función.

Distancia de Manhattan en el juego del 8

4	1	3
2		5
7	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

$$h_1(\text{Inicial}, \text{Final}) = 1 + 1 + 1 + 1 + 1 = 5$$

$$h_2(\text{Inicial}, \text{Final}) = 1 + 1 + 2 + 1 + 1 = 6$$

h_2 es una mejor aproximación.

Distancia de Manhattan en el juego del 8

 <p>Inicial</p>	 <p>Final</p>
$h_1(\text{Inicial}, \text{Final}) = 1 + 1 + 1 + 1 + 1 = 5$ $h_2(\text{Inicial}, \text{Final}) = 1 + 1 + 2 + 1 + 1 = 6$ h_2 es una mejor aproximación.	

Ambas heurísticas no sobreestiman, esto quiere decir que las estimaciones se mantienen por debajo del costo real.

La más cercana al costo real debe ser la que nos entregue un valor más grande.

En este caso la distancia de Manhattan nos da una mejor aproximación del costo estimado para llegar a la meta.

Distancia de Manhattan en el juego del 8

4	1	3
2		5
7	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

Si tenemos n heurísticas h_1, h_2, \dots, h_n
podemos combinarlas en una nueva:
$$h_{\text{combinada}}(s) = \max\{h_1(s), h_2(s), \dots, h_n(s)\}$$

Distancia de Manhattan en el juego del 8



Inicial



Final

Si tenemos n heurísticas h_1, h_2, \dots, h_n
podemos combinarlas en una nueva:
$$h_{\text{combinada}}(s) = \max\{h_1(s), h_2(s), \dots, h_n(s)\}$$

Esto nos permite plantear una manera de incorporar varias heurísticas en una nueva heurística más informativa.

Si tenemos n heurísticas distintas, todas ellas admisibles, proponemos una heurística nueva que calcula el máximo de los valores resultantes de aplicar cada heurística a un estado.

Distancia de Manhattan en el juego del 8

4	1	3
2		5
7	8	6

Inicial

1	2	3
4	5	6
7	8	

Final

Sin embargo para este ejemplo,
tenemos que:

$$\forall s. h_2(s) \geq h_1(s)$$

por lo que h_2 es suficiente.

Distancia de Manhattan en el juego del 8



Inicial



Final

Sin embargo para este ejemplo,
tenemos que:
 $\forall s. h_2(s) \geq h_1(s)$
por lo que h_2 es suficiente.

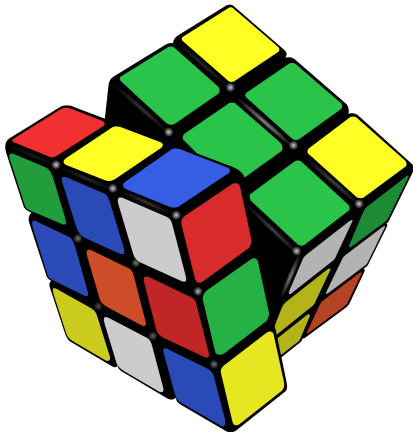
Pero hay que tener cuidado.

En este ejemplo es claro que la heurística h_2 siempre nos entregará un estimado mayor o igual al de la heurística h_1 .

Por lo tanto, es infructuoso calcular ambas.

Nos quedamos con la distancia de Manhattan.

Heurísticas basadas en base de datos de patrones

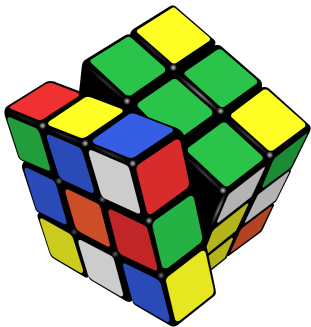


└─ Heurísticas basadas en base de datos de patrones



A veces encontrar heurísticas basadas en distancias no es suficiente.

Heurísticas basadas en base de datos de patrones



Es posible calcular una distancia de Manhattan en 3 dimensiones para un cubo de Rubik.
Pero esta es una heurstica poco informativa.

└─ Heurísticas basadas en base de datos de patrones



Es posible calcular una distancia de Manhattan en 3 dimensiones para un cubo de Rubik.
Pero esta es una heurstica poco informativa.

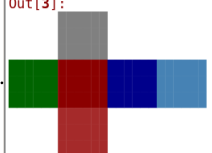
Para el cubo de Rubik, es posible construir una heurística basada en una distancia de Manhattan en 3 dimensiones.

Sin embargo, esta heurística es poco informativa.

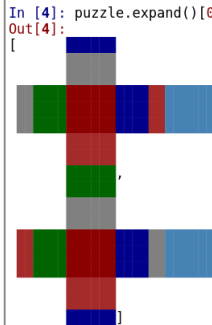
Heurísticas basadas en base de datos de patrones

Partiendo del cubo ordenado.
Si rotamos una cara 90 grados:
Se desacomodan 8 subcubos simultáneamente.
Una distancia contando subcubos sería
no admisible.

```
In [3]: puzzle
Out[3]:
```



```
In [4]: puzzle.expand()[0:2]
Out[4]:
```



Heurísticas basadas en base de datos de patrones

Para observar la razón, veamos que sucede si aplicamos una acción al cubo ordenado.

Digamos que lo rotamos 90 grados.

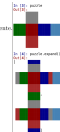
En esta figura vemos del lado derecho dos configuraciones posibles si rotamos el cubo una vez.

Hemos aplanado las caras para poder verlas simultáneamente. Hay que imaginar que el cubo se obtiene doblando 90 grados en cada unión de las caras.

Observamos que se desacomodan 8 subcubos.

Si tomáramos de manera simple el número de subcubos fuera de lugar, esto resultaría en una heurística que sobreestima el valor real. En este caso solo se realizó un movimiento. Es decir tenemos un costo de real de 1.

Partiendo del cubo ordenado.
Si rotamos una cara 90 grados:
Se desacomodan 8 subcubos simultáneamente.
Una distancia contando subcubos sería
no admisible.

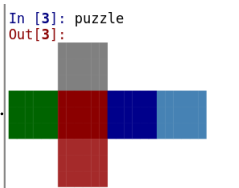


Heurísticas basadas en base de datos de patrones

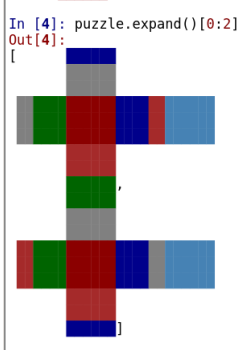
Partiendo del cubo ordenado.
Si rotamos una cara 90 grados:
Se desacomodan 8 subcubos simultáneamente.
Una distancia contando subcubos sería
no admisible.

La distancia de Manhattan
tiene que dividirse entre 8.

```
In [3]: puzzle
Out[3]:
```



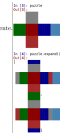
```
In [4]: puzzle.expand()[0:2]
Out[4]:
```



Heurísticas basadas en base de datos de patrones

Partiendo del cubo ordenado.
Si rotamos una cara 90 grados:
Se desacomodan 8 subcubos simultáneamente.
Una distancia contando subcubos sería
no admisible.

La distancia de Manhattan
tiene que dividirse entre 8.



Para hacer la distancia de Manhattan en tres dimensiones una
heurística admisible se tendría que dividir el valor entre 8.

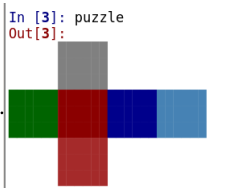
Heurísticas basadas en base de datos de patrones

Partiendo del cubo ordenado.
Si rotamos una cara 90 grados:
Se desacomodan 8 subcubos simultáneamente.
Una distancia contando subcubos sería
no admisible.

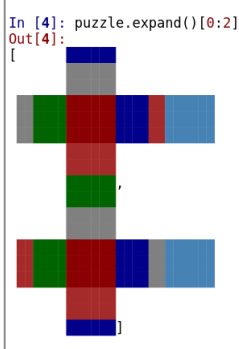
La distancia de Manhattan
tiene que dividirse entre 8.

Esta heurística no es buena.
La distancia real puede ser de
hasta 20 movimientos cuando se
consideran giros de 90 y 180
grados en las caras.

```
In [3]: puzzle
Out[3]:
```



```
In [4]: puzzle.expand()[0:2]
Out[4]:
```



Heurísticas basadas en base de datos de patrones

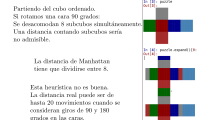
Con ayuda de las supercomputadoras de Google Tomas Rokicki demostró en 2010 que la distancia máxima a la que pueden estar dos configuraciones del cubo es 20 cuando se consideran como acciones rotaciones de las caras de 90 y 180 grados.

Si tenemos un total de 26 subcubos, de los cuales los seis cubos centrales de cada cara no pueden moverse. Podemos ver que 20 entre 8 nos daría una distancia máxima de 2.5.

Esta heurística es muy mala, pues esta muy alejada de el valor máximo esperado que es 20.

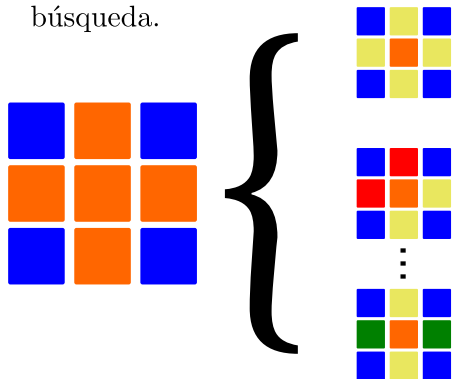
Korf propone tomar solo algunos subcubos por ejemplo los de las esquinas o los de las orillas en las cruces.

Tomando los de las orillas la distancia Manhattan se divide entre 4 y el valor esperado de distancia es de 5.5. Esto sigue siendo malo para guiar a los algoritmos informados en un espacio de más de 43 trillones de configuraciones.



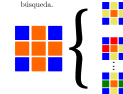
Heurísticas basadas en base de datos de patrones

Una base de datos de patrones (Korf 97; Culberson y Schaeffer 96), es una manera alternativa de obtener información para guiar la búsqueda.



Heurísticas basadas en base de datos de patrones

Una base de datos de patrones (Korf 97; Culberson y Schaeffer 96), es una manera alternativa de obtener información para guiar la búsqueda.



Korf propone el uso de bases de datos de patrones, una idea desarrollada por Culberson y Schaeffer en 1996 para el rompecabezas del 15.

La idea es simple, partiendo de la configuración meta. Hay que realizar una búsqueda tipo BFS y observar cada configuración descubierta y poner atención solo a un subconjunto de los subcubos.

Por ejemplo, aquí ponemos atención solo a los subcubos de las esquinas.

Guardamos en una base de datos el patrón descubierto junto con la profundidad mínima a la que se encuentra dicho patrón.

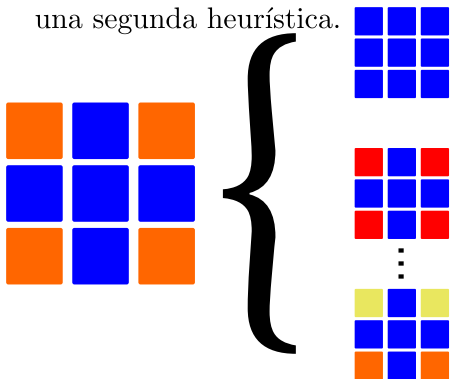
Como vemos en la figura el patrón con los subcubos azules en las esquinas captura todos los estados en los que esto es posible.

Las configuraciones mostradas todas tienen el mismo patrón de esquinas.

La configuración a la que se encontró el patrón de esquinas azules, sirve como heurística para todos los demás estados.

Heurísticas basadas en base de datos de patrones

Korf utilizó los cubos de las esquinas para generar una heurística y los cubos de las cruces para generar una segunda heurística.



Heurísticas basadas en base de datos de patrones

Korf utilizó los cubos de las esquinas para generar una heurística y los cubos de las cruces para generar una segunda heurística.



Aquí mostramos un segundo patrón. Ahora nos fijamos en los cubos de las orillas, los que forman la cruz.

Al descubrirse este patrón de orillas azules, el algoritmo tipo BFS guardará el patrón junto con su profundidad.

La profundidad servirá para todo estado que exhiba el patrón, un conjunto muy grande.

Por construcción esta heurística es admisible.

Conclusión

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.

Conclusión

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.

Sumarizando:

podemos relajar las restricciones del problema y proponer una heurística basada en una distancia entre estados.

Conclusión

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.

Conclusión

Se pueden combinar más de una heurística admisibles utilizando la función máximo sobre los valores calculados con cada una de ellas.

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.

Conclusión

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.
- Hay que poder evaluar las heurísticas de manera eficiente.

Conclusión

Las heurísticas deben poder calcularse de manera rápida, de otra forma minarán el desempeño de los algoritmos pues son invocadas con cada expansión de estados.

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.
- Hay que poder evaluar las heurísticas de manera eficiente.

Conclusión

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.
- Hay que poder evaluar las heurísticas de manera eficiente.
- Podemos generar estimaciones de costos mínimos de un conjunto de estados descrito por un patrón al estado meta y almacenarlos en una base de datos.

Conclusión

Con el uso de bases de datos de patrones se pueden generar heurísticas fuera de línea y ser guardadas en bases de datos.

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.
- Hay que poder evaluar las heurísticas de manera eficiente.
- Podemos generar estimaciones de costos mínimos de un conjunto de estados descrito por un patrón al estado meta y almacenarlos en una base de datos.

Conclusión

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.
- Hay que poder evaluar las heurísticas de manera eficiente.
- Podemos generar estimaciones de costos mínimos de un conjunto de estados descrito por un patrón al estado meta y almacenarlos en una base de datos.
- Los agentes pueden incorporar técnicas de aprendizaje máquina para aprender las heurísticas (Arfaee et al. 2011).

Conclusión

Existe investigación reciente sobre cómo un agente podría usar aprendizaje máquina para proponer heurísticas de manera automática.

Por ejemplo Arfaee, Zilles y Holte, aplican técnicas de aprendizaje máquina en el cubo de Rubik para mejorar las heurísticas de Korf basadas en bases de datos de patrones.

- Típicamente podemos construir una función heurística relajando las restricciones del problema original y planteando una distancia entre configuraciones.
- Podemos combinar varias heurísticas en una sola, siempre que nos den información distinta.
- Hay que poder evaluar las heurísticas de manera eficiente.
- Podemos generar estimaciones de costos mínimos de un conjunto de estados descrito por un patrón al estado meta y almacenarlos en una base de datos.
- Los agentes pueden incorporar técnicas de aprendizaje máquina para aprender las heurísticas (Arfaee et al. 2011).